

Distributed Chance-Constrained Task Allocation for Autonomous Multi-Agent Teams

Sameera S. Ponda, Luke B. Johnson and Jonathan P. How

Abstract—This research presents a distributed chance-constrained task allocation framework that can be used to plan for multi-agent networked teams operating in stochastic and dynamic environments. The algorithm employs an approximation strategy to convert centralized problem formulations into distributable sub-problems that can be solved by individual agents. A key component of the distributed approximation is a risk adjustment method that allocates individual agent risks based on a global risk threshold. The results show large improvements in distributed stochastic environments by explicitly accounting for uncertainty propagation during the task allocation process.

I. INTRODUCTION

The use of autonomous robotic agents for complex missions, such as unmanned aerial vehicles (UAVs) and ground rovers, has motivated the development of autonomous task allocation and planning methods. The goal of such algorithms is to distribute the mission tasks amongst the agents to ensure spatial and temporal coordination of the team. The basic problem can be formulated as a combinatorial optimization mixed-integer program, involving nonlinear and time-varying system dynamics. For most problems of interest, optimal solution methods are computationally intractable and approximation techniques are regularly employed [1]. Centralized planning approaches usually require high bandwidth, are resource intensive, and react slowly to changes in dynamic environments, motivating the development of distributed algorithms where agents coordinate with each other locally [2]. One class of distributed planning algorithms involves using auction algorithms augmented with consensus protocols, which are particularly well suited to developing real-time conflict-free solutions [2]–[5].

An important issue associated with planning for heterogeneous networked teams is that planning algorithms rely on underlying system models and parameters, which are usually approximations of the real systems involving many limiting assumptions and complex dynamics. Discrepancies between these planner models and the actual system dynamics cause degradations in mission performance. Furthermore, the impact of these discrepancies on the overall quality of the plan is typically hard to quantify in advance due to nonlinear effects, coupling between tasks and agents, and interdependencies between system constraints (e.g. longer-than-expected service times impact the arrival times of subsequent

tasks). However, if uncertainty models of planning parameters are available they can be leveraged to create robust plans that explicitly hedge against the inherent uncertainty.

Such stochastic planning methods have been employed for several types of optimization problems, ranging from robust portfolio optimization strategies [6] to UAV planning [3], to robust scheduling for airline operations [7]. While these varied approaches provide valuable insights, they also highlight several key challenges associated with robust planning, including coupling probability distributions through scoring and constraint functions, evaluating stochastic metrics, and maintaining computational tractability and solution scalability. Furthermore, inherent assumptions adopted in most of the robust planning literature (e.g. linearity, specific homogeneous i.i.d. distributions), limit the applicability of these approaches when planning for more general realistic mission scenarios. This paper addresses these challenges by developing distributed stochastic planning strategies that can effectively embed uncertainty models of planning parameters into the score functions, transition dynamics, and constraints, producing real-time approximate solutions to the task allocation problem for networked multi-agent teams.

II. PROBLEM STATEMENT

A. Problem Formulation

Given a list of N_a agents and N_t tasks, the goal of the task allocation algorithm is to find a conflict-free matching of tasks to agents that maximizes the global reward¹. The objective function for the mission is given by a sum over local objective functions for each agent, which are in turn functions of the tasks assigned to that agent, the times at which those tasks will be executed, and the set of planning parameters. This task assignment problem can be written as the following mixed-integer (possibly nonlinear) program:

$$\begin{aligned} \max_{\mathbf{x}, \boldsymbol{\tau}} \quad & \sum_{i=1}^{N_a} \sum_{j=1}^{N_t} c_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) x_{ij} \\ \text{s.t.} \quad & \mathbf{G}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) \leq \mathbf{b} \\ & \mathbf{x} \in \{0, 1\}^{N_a \times N_t}, \boldsymbol{\tau} \in \{\mathbb{R}^+ \cup \emptyset\}^{N_a \times N_t} \end{aligned} \quad (1)$$

where \mathbf{x} , is a set of binary decision variables, x_{ij} , which are used to indicate whether or not task j is assigned to agent i ; $\boldsymbol{\tau}$ is the set of real-positive decision variables τ_{ij} indicating when agent i will execute its assigned task j ; $\boldsymbol{\theta}$ is the set of planning parameters that affect the score

S. Ponda and L. Johnson are with the Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, {sponda, lbj16}@mit.edu

J. P. How is the R. C. Maclaurin Professor of Aeronautics and Astronautics, MIT, Cambridge, MA, jhow@mit.edu

¹An assignment is said to be conflict-free if each task is assigned to no more than one agent.

calculation; c_{ij} is the reward agent i receives for task j given the overall assignment and parameters; and $\mathbf{G} = [\mathbf{g}_1 \dots \mathbf{g}_{N_c}]^T$, with $\mathbf{b} = [b_1 \dots b_{N_c}]^T$, is a set of N_c possibly nonlinear constraints of the form $\mathbf{g}_k(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) \leq b_k$ that capture transition dynamics, resource limitations, etc. This generalized problem formulation can accommodate several different design objectives and constraints commonly used in multi-agent decision making problems (e.g. search and surveillance missions where c_{ij} represents the value of acquired information and the constraints \mathbf{g}_k capture fuel limitations, or rescue operations where c_{ij} is time-critical favoring earlier τ_{ij} execution times).

An important observation is that, in Eq. (1), the scoring and constraint functions are explicitly dependent upon the decision variables \mathbf{x} and $\boldsymbol{\tau}$, which makes this general mixed-integer programming problem very difficult to solve (NP-hard) due to the inherent system inter-dependencies [1]. To make matters worse, in realistic mission scenarios the planning parameters are typically a combination of deterministic and stochastic variables, and the above optimization must account for the uncertainty in $\boldsymbol{\theta}$, increasing the dimensionality of the problem and further exacerbating computational intractability [8].

B. Stochastic Planning Framework

Assume that a model of the uncertainty is available, where $\boldsymbol{\theta} \in \Theta$ and is distributed according to the probability density function (PDF), $f(\boldsymbol{\theta})$. The goal of the stochastic planner is to use the information provided in $f(\boldsymbol{\theta})$ to create plans that account for the *variability* of $\boldsymbol{\theta}$. There are several metrics that can be used to account for uncertainty in the planning formulation. Perhaps the most common approach is to maximize the expected mission performance [9], where the objective function from Eq. (1) becomes

$$\max_{\mathbf{x}, \boldsymbol{\tau}} \mathbb{E}_{\boldsymbol{\theta}} \left(\sum_{i=1}^{N_a} \sum_{j=1}^{N_t} c_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) x_{ij} \right) \quad (2)$$

Note that optimizing Eq. (2) is *not* the same as planning using the mean values of $\boldsymbol{\theta}$. In fact, planning using mean parameter values often leads to poor planning performance, since the problem formulation fails to capture the non-trivial coupling of uncertainty in scores, dynamics and constraints. This is especially problematic when scores are coupled, and can lead to biased predictions that drastically misrepresent the actual expected performance.

While optimizing Eq. (2) provides a plan that maximizes the expected performance of the system, an actual single run execution of this best expected plan is still subject to the uncertainty in the environment, and may result in a relatively poor plan (worse than expected) with some non-zero probability. If the current mission tolerance to failure is very low, a more *robust* planning objective is to maximize the worst case scenario,

$$\max_{\mathbf{x}, \boldsymbol{\tau}} \min_{\boldsymbol{\theta}} \left(\sum_{i=1}^{N_a} \sum_{j=1}^{N_t} c_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) x_{ij} \right) \quad (3)$$

Optimizing Eq. (3) ensures that the plan execution will result in a score *no worse* than that predicted by the algorithm. However, these classical robust formulations that mitigate the worst case system performance [8,10] are usually too conservative, motivating the development of stochastic planning methods that find the best solution within some predefined risk threshold [11,12].

The fields of Robust Optimization and Stochastic Optimization [12] have addressed this issue of conservatism through several different methods. One such approach involves optimizing a risk-adjusted expected performance, where a risk function $R(c_{ij})$ biases the original cost function c_{ij} towards more conservative solutions to account for the acceptable level of risk. Another approach is to bound the domain of the uncertainty set $\boldsymbol{\theta}$ to be within certain ranges, $\boldsymbol{\theta} \in [\Theta_{\min}, \Theta_{\max}] \subset \Theta$, or to take on a set of discrete representative values $\boldsymbol{\theta} \in [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_k] \subset \Theta$, thus limiting the support of the uncertain parameters. Classical robust convex optimization techniques can then be used to solve the resulting approximate problem (examples of these approaches include bounding uncertainty using ellipsoids for specified confidence intervals [13], constructing representative uncertainty sets [14,15], etc.). Although these methods allow control over the planner risk, there are a few issues which makes their practical implementation difficult. Firstly, it is not usually obvious how to design the risk function $R(c_{ij})$ or the bounded sets for $\boldsymbol{\theta}$, and the selection of these is typically problem specific, time consuming and ad-hoc. A more serious issue, however, is that the metric of interest is the *cumulative* risk of the total solution, not the individual parameter or task risks, and it is difficult to quantify how these individual parameter bounds will affect the global risk. Nonlinearities in the cost functions, complex variable coupling and interdependencies, and discrete optimization effects, often affect the solution in unpredictable ways, and it is therefore hard to ensure that the total mission outcome is within the desired risk threshold.

An alternative formulation that guarantees that the global mission performance will be within a certain risk threshold is the chance-constrained formulation [11,16,17],

$$\begin{aligned} \max_{\mathbf{x}, \boldsymbol{\tau}} \quad & y \\ \text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\theta}} \left(\left(\sum_{i=1}^{N_a} \sum_{j=1}^{N_t} c_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) x_{ij} \right) > y \right) \geq 1 - \epsilon \end{aligned} \quad (4)$$

The goal of Eq. (4) is to maximize the worst-case score within the allowable risk threshold specified by ϵ (can be interpreted as guaranteeing that solution will be at least as good as y with probability greater than or equal to $1 - \epsilon$). When $\epsilon = 0$, the score is guaranteed to be at least y with probability one (absolute worst-case), and the chance-constrained formulation reduces to the robust formulation of Eq. (3). The main drawback of the chance-constrained formulation is that it is difficult to solve, especially given the extensive coupling between agents and tasks (double sum over distributions). Previous work has mainly considered

linear or quadratic optimization with continuous variables [11,16,17], where, under special circumstances, optimal solutions and bounds can be found analytically. However, task allocation is a mixed-integer program, and these techniques cannot be easily extended to discrete optimization, especially given nonlinear and heterogeneous score functions with coupled distributions. Furthermore, these solution strategies are centralized and cannot be trivially extended to distributed environments. This work addresses these issues by developing an efficient approximation algorithm that provides *distributed* numerical solutions to the chance-constrained task allocation problem.

III. CHANCE-CONSTRAINED TASK ALLOCATION

The chance-constrained task allocation problem can be written as,

$$\begin{aligned} \max_{\mathbf{x}, \boldsymbol{\tau}} \quad & y \tag{5} \\ \text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\theta}} \left(\left(\sum_{i=1}^{N_a} \sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) x_{ij} \right) > y \right) \geq 1 - \epsilon \\ & \sum_{i=1}^{N_a} x_{ij} \leq 1, \forall j; \mathbf{x} \in \{0, 1\}^{N_a \times N_t}, \boldsymbol{\tau} \in \{\mathbb{R}^+ \cup \emptyset\}^{N_a \times N_t} \end{aligned}$$

For simplicity, the only hard constraint assumed in this problem formulation is that no task can be assigned to more than one agent². It is of interest to consider how the parameter uncertainty impacts these task execution times and how coupling between task arrival times affects the performance of the mission. Consider a multi-agent multi-task UAV mission, where agents must perform time-critical target search and track tasks, but where task service times and agent travel velocities are uncertain. The score functions are specified by

$$\mathbf{c}_{ij}(\tau_{ij}) = \begin{cases} R_j e^{-\lambda_j \Delta \tau_{ij}}, & (t_{j_{\text{start}}} + \bar{t}_{j_{\text{duration}}}) \leq \tau_{ij} \leq t_{j_{\text{end}}} \\ 0, & \text{otherwise.} \end{cases}$$

where the task time-window $[t_{j_{\text{start}}}, t_{j_{\text{end}}}]$ represents the period of time in which the task must be completed, τ_{ij} is the time at which agent i finishes executing task j , and $\Delta \tau_{ij} = \tau_{ij} - (t_{j_{\text{start}}} + \bar{t}_{j_{\text{duration}}})$ represents the time in excess of the expected task completion time. The exponential decay represents the time-critical nature of the task, where the discount factor λ_j is used to reduce the nominal reward R_j . The arrival time at task j is a function of the agent's assignment, its other task arrival times, and the uncertain task durations, $\tau_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta})$.

There are several issues that make this problem particularly hard to solve. Firstly, the tasks are temporally coupled (stochastic task durations and travel velocities for early tasks affect arrival times and thus scores for later tasks). Given a set of tasks, finding a PDF involves computing the distribution of a sum of non-independent heterogeneous task

scores. This expression is analytically intractable, and even for simple i.i.d. distributions involves computing the convolution over all tasks in the bundle (only tractable for special cases: Gaussian distributions, exponential-Erlang, etc.). In general, numerical methods can be employed to approximate the combined distribution via particle simulation. The second issue involves developing algorithms to select the best task sets. Even in a deterministic setting, this problem is NP-hard and involves enumerating possible bundle combinations. Furthermore, the problem formulation involves selecting the optimal task times, as well as dealing with the uncertain parameters. Since optimal solutions are intractable, good approximate solutions can be developed that incrementally build the task sets (such as sequential greedy). The main challenge in this work is to ensure that the goal of optimizing the percentile score given the *total* risk for the fleet is appropriately represented in the sequential process. Finally, when planning for multi-agent teams, distributed planning strategies can offer advantages. The challenge with this problem involves developing expressions that relate each agent's local risk to the global risk within a theoretically sound framework. Given these individual risk allotments, agents can perform distributed planning to optimize their own plans. Expressions for choosing each agent's risk are analytically intractable and problem specific, so the challenge lies in developing good approximations to relating the global and local risk thresholds.

IV. DISTRIBUTED APPROXIMATION TO THE CHANCE-CONSTRAINED TASK ALLOCATION PROBLEM

This section presents a distributed approximation to the above problem formulation, and discusses algorithmic strategies to solve this distributed problem..

A. Individual Agent Problem Statement

The problem presented in Eq. (5) can be decomposed into distributable sub-problems by the approximation,

$$\begin{aligned} \max_{\mathbf{x}, \boldsymbol{\tau}} \quad & \sum_{i=1}^{N_a} y_i \tag{6} \\ \text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\theta}} \left(\left(\sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) x_{ij} \right) > y_i \right) \geq 1 - \epsilon_i, \forall i \\ & \sum_{i=1}^{N_a} x_{ij} \leq 1, \forall j; \mathbf{x} \in \{0, 1\}^{N_a \times N_t}, \boldsymbol{\tau} \in \{\mathbb{R}^+ \cup \emptyset\}^{N_a \times N_t} \end{aligned}$$

where each agent i solves its own chance-constrained optimization to maximize y_i subject to its individual risk threshold ϵ_i , while ensuring, through communication with other agents, that the joint constraint for a non-conflicting solution remains satisfied. Note that ϵ_i is not the same as the original mission risk ϵ (more details about this in the next section).

This work employs the Consensus-Based Bundle Algorithm (CBBA) to solve Eq. (6) (see [2,3,5] for further details). Given a well synchronized distributed system, this

²Not addressed in this paper is uncertainty in constraints which is typically much harder to deal with. This involves ensuring that constraints are satisfied within a given probability, e.g. $\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{G}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) \leq \mathbf{b}) > \alpha$.

algorithm can be quite fast, and converges in polynomial time with respect to tasks and agents. The greedy process by which each agent builds its task bundle in CBBA needs to ensure that for every bid, the task scores satisfy the probabilistic constraints specified by Eq. (6). The first challenge is to represent the actual score distribution of a task that is being added to the bundle. This is done by sampling all of the random variables that could possibly affect the value added from this task (in this case the random variables are the task durations and agent velocities). Using these samples the marginal benefit of introducing this task into the bundle is computed. Even though the process involves placing the task in the ideal spot in continuous time, given the decaying cost functions, there are only a discrete number of continuous times that could maximize this sample's score for the task. This entire process is repeated N times, where N is a sampling value large enough to create a representative probability distribution of the scores obtained from adding this task. Given this distribution, the bid for the particular task becomes the $N\epsilon_i^{\text{th}}$ lowest score particle (worst score within allowable risk). If the distribution is sufficiently represented then this choice guarantees that the score of the bid will be truly at least this large with probability $1 - \epsilon_i$. The algorithm cycles through all possible tasks and picks the one with the highest score to add to the bundle. This process repeats until the agent cannot add any additional tasks, then he is ready to communicate with others. Note that this process of adding tasks to the bundle satisfies the true ϵ_i for the agent each time a task is added, so the algorithm remains consistent with the equations presented in Eq. (6).

Although this decomposition makes the problem easier to solve in a distributed fashion, it also introduces the additional complexity of picking the parameters ϵ_i such that the goal of maximizing the chance-constrained score of the mission distribution, $y = \sum_{i=1}^{N_a} y_i$, given the mission risk ϵ , is adequately represented. For generic task allocations, the relationship between the mission ϵ and each of the agents' ϵ_i 's is non-trivial, however, given certain probability models, the complexity of picking these values can be reduced (note that doing this efficiently is still an open research question). This work addresses these issues by employing an approximation strategy that attempts to model the individual agents' risks as a function of the global mission risk. This *risk adjustment* method invokes the Central Limit Theorem and uses Gaussian approximations of the distributions to develop an expression for agent risk as a function of mission risk. The next section provides details on how this is accomplished.

B. Risk Adjustment for Distributed Approximation

Given a mission risk value ϵ , the constraint in the original problem (Eq. (5)) becomes tight for y values which satisfy the probability constraint with equality. Using the cumulative distribution function (CDF) notation for any random variable X , $\mathbb{P}_X(X \leq x) = F_X(x)$, the tight constraint from Eq. (5)

can be written as,

$$\mathbb{P}_{\theta} \left(\sum_{i=1}^{N_a} y_i > y \right) = 1 - \epsilon \Rightarrow F_{\sum y_i}(y) = \epsilon$$

where $y_i = \sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}, \boldsymbol{\tau}, \boldsymbol{\theta}) x_{ij}$. Similarly, when the individual agent sub-problem constraints are tight, the constraint equations become,

$$\mathbb{P}_{\theta} \left(\sum_{j=1}^{N_t} \mathbf{c}_{ij} x_{ij} > y_i \right) = 1 - \epsilon_i \Rightarrow F_{y_i}(y_i) = \epsilon_i \quad \forall i$$

If the CDFs are assumed invertible (e.g. continuous differentiable distributions), then combining the above expressions establishes a relationship between $\epsilon = F_{\sum y_i}(y)$ and ϵ_i expressed as

$$\epsilon = F_{\sum y_i} \left(\sum_{i=1}^{N_a} y_i \right) = F_{\sum y_i} \left(\sum_{i=1}^{N_a} F_{y_i}^{-1}(\epsilon_i) \right) \quad (7)$$

What Eq. (7) implies is that, if individual sub-problems are solved that predict scores y_i given risk values ϵ_i , and if these predicted scores are added to obtain a global score y , this will equate to a mission risk value ϵ , as specified by Eq. (7), given CDF expressions for the individual and global distributions³. This provides a metric to evaluate how closely the distributed approximation accomplishes the original centralized problem objective.

The risk adjustment method employed in this work uses the expression provided by Eq. (7), along with a Gaussian assumption for the distributions, to develop an expression for the agent risk values ϵ_i in terms of the global mission risk ϵ . Note that, for general problems, Eq. (7) will give infinite possible combinations of ϵ_i given a specific value of ϵ , as long as the equation is satisfied. To avoid this issue, it is assumed that all agents are given identical risk values ϵ_i (note that this does *not* imply that the agents have identical distributions). If the agent distributions are Gaussian, $y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, then the mission distribution will also be Gaussian with mean and variance given by, $y \sim \mathcal{N}(\sum_{i=1}^{N_a} \mu_i, \sum_{i=1}^{N_a} \sigma_i^2)$. As a reminder, the CDF and inverse CDF expressions for a Gaussian distribution are given by,

$$\begin{aligned} F_x(x) &= \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{x - \mu}{\sqrt{2\sigma^2}} \right) \right) \\ F_x^{-1}(x) &= \mu + \sqrt{2\sigma^2} \operatorname{erf}^{-1}(2x - 1) \end{aligned}$$

Using these equations in combination with Eq. (7), and assuming identical agent risk values, the following expression is obtained,

$$\epsilon_i = \frac{1}{2} \left(1 + \operatorname{erf} \left(C \operatorname{erf}^{-1}(2\epsilon - 1) \right) \right) \quad (8)$$

³Note that these CDF expressions are often intractable given that the PDF of the mission distribution involves a convolution of individual agent distributions, and likewise the individual agent distributions involve combinations and couplings of task distributions.

$$C = \frac{\sqrt{\sum_{i=1}^{N_a} \sigma_i^2}}{\sum_{i=1}^{N_a} \sqrt{\sigma_i^2}} \quad (9)$$

This expression has several interesting properties. Firstly, the agent risk values for the Gaussian case do not depend on the means of the agent distributions or mission distribution, they only depend on the variances. Also, if the agents are homogeneous (identical distributions), then $C = 1/\sqrt{N_a}$, and if the entire mission distribution comes from only 1 agent's contribution, then $C = 1$ and $\epsilon_i = \epsilon$ as expected. This implies that team heterogeneity can be represented via the factor C , which is dependent on N_a . Eq. (9) provides the risk adjustment method used by the distributed algorithm given the number of agents and a measure of team heterogeneity.

V. RESULTS AND DISCUSSION

The distributed chance-constrained task allocation algorithm was implemented in simulation and tested on a UAV mission as described in the previous section. The uncertainty models for all random variables were represented by log-normal distributions. There were two types of tasks: high-reward high-uncertainty tasks, and lower reward tasks with low variance. The UAV team was also composed of two types of agents: fast but unpredictable agents (high mean and high variance), and slower speed but more predictable (low mean and low variance).

A Monte Carlo simulation was performed and the results are presented in Figure 1. The left set of plots, Figures 1(a) and 1(d), show the results for a single agent mission with 10 tasks. The top plot shows a comparison between the deterministic planner, the robust planner ($\epsilon = 0$) and the chance-constrained planner for different levels of risk (with a sequential greedy centralized planner shown for comparison as the black dotted line). As a reminder, 5% allowable risk means that with 95% probability the planner will get at least the predicted score. Since the chance-constrained planner is aware that this is the metric to optimize, it can achieve higher guaranteed scores than the other two methods for any level of risk. The robust planner also achieves higher scores than the deterministic planner for low risk values, and is equal to the chance-constrained planner at 0% risk, but as the amount of allowable risk increases the robust solution is too conservative and the benefit over deterministic planning starts to diminish. The chance-constrained planner, however, considers this additional risk margin and chooses less conservative (but higher performing) plans. The main intuition as to why the chance-constrained planner outperforms the deterministic approach is that the deterministic planner does not account for the variances in the task service times and travel velocities, and thus tries to squeeze tasks into the current path based on expected values of these parameters. The stochastic planner on the other hand accounts for this variance and will not add tasks to the path if they impact high value tasks (with significant probability). The overall result is that chance-constrained task allocation reduces the variance in performance. Figure 1(d), which shows a histogram of the planner distributions, illustrates this effect. The particular

histograms are for a simulation run with a 5% risk threshold. As shown in the figure, the robust and chance-constrained distributions are much narrower than the deterministic one, which is more spread out (larger tail on the distribution). The 5% risk scores (shown as dotted lines) indicate that the chance-constrained score is higher than the other two. An interesting thing to note in Figure 1(d) is that the distributions are mixed continuous and discrete⁴, which is one of the many reasons that traditional robust optimization methods are not amenable to these types of problems.

The second set of plots, Figures 1(b) and 1(e), show the results for a 6 agent mission with 60 tasks, which look very similar to the single agent case. Once again, the chance-constrained planner outperforms the other two approaches and the robust planner performs higher than deterministic at low risk values. The main things to note in this multi-agent scenario are that the distributed chance-constrained planner achieves equivalent performance to the centralized sequential greedy approach, thus validating the distributed approximation to the centralized chance-constrained problem; and that the distributions for the multi-agent mission scores are nearly Gaussian, justifying the use of the Gaussian approximation in the risk adjustment method. In fact, as the number of agents and tasks increases to infinity, the mission distributions tend to Gaussian as predicted by the Central Limit Theorem. The risk adjustment method used in this scenario assumed a heterogeneous team with two types of agents (the team heterogeneity factor was set to $C = \sqrt{2/N_a}$ in Equation (9)).

The last set of plots, Figures 1(c) and 1(f), show the benefit of adjusting the risk for individual agents given a global mission risk. Figure 1(f) shows the achieved mission risk due to the distributed approximation (as described in Equation (7)) for the algorithm with and without using the risk adjustment method. The dotted line on the plot represents a perfect match between desired and actual mission risk. As shown in the figure, without risk adjustment the team performs conservatively, achieving a much lower mission risk than allowed and thus sacrificing performance. With the risk adjustment method, the team is able to more accurately predict the mission risk, achieving higher performing plans within the allowable threshold. This is illustrated in Figure 1(c), which shows that the distributed algorithm with risk adjustment consistently achieves higher performance and is on par with the centralized sequential greedy algorithm.

VI. CONCLUSION

This paper presents a distributed chance-constrained task allocation framework that can be used to plan for multi-agent networked teams operating in stochastic and dynamic environments. The algorithm employs an approximation strategy that decomposes the centralized problem into distributable sub-problems which can be solved by individual agents. A risk adjustment method is presented that enhances the

⁴The reason for this is that task execution times outside the task time-windows of validity result in zero score, which produces discontinuities and discrete jumps in the score distributions.

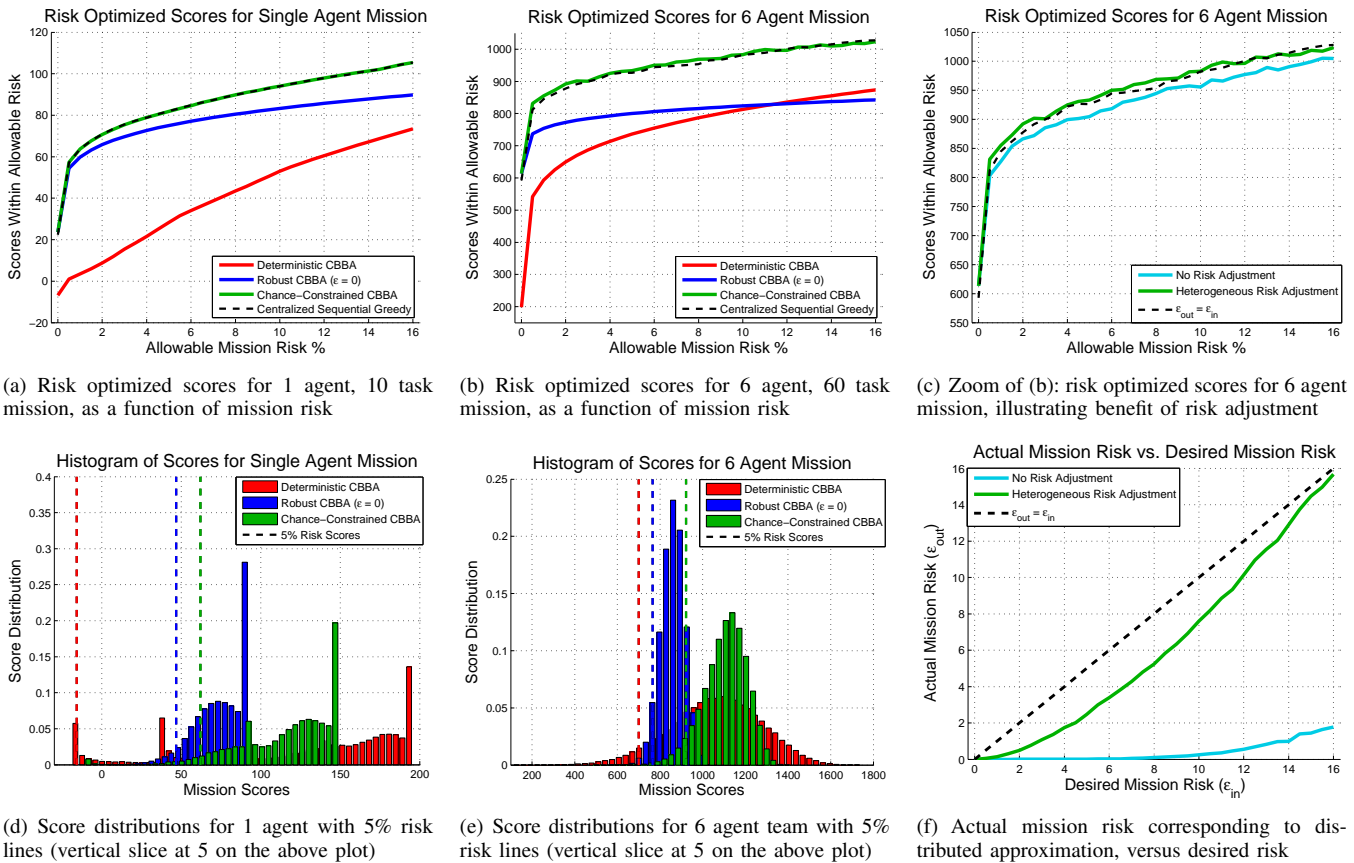


Fig. 1. Monte Carlo simulation results for multi-agent multi-task stochastic missions illustrating the benefits of distributed chance-constrained planning over deterministic and robust planning strategies.

distributed approximation by allocating individual agent risks based on a global risk threshold. The results show that by explicitly accounting for uncertainty propagation during the task allocation process large improvements can be made for distributed multi-agent teams operating in stochastic environments.

ACKNOWLEDGMENTS

This research was supported in part by AFOSR (FA9550-08-1-0086) and MURI (FA9550-08-1-0356). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

REFERENCES

- [1] D. Bertsimas and R. Weismantel, *Optimization over integers*. Dynamic Ideas Belmont, MA, 2005.
- [2] H. Choi, L. Brunet, and J. How, "Consensus-based decentralized auctions for robust task allocation," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 912–926, 2009.
- [3] S. Ponda, J. Redding, H.-L. Choi, J. P. How, M. A. Vavrina, and J. Vian, "Decentralized planning for complex missions with dynamic communication constraints," in *American Control Conference (ACC)*, Baltimore, MD, July 2010.
- [4] L. Johnson, "Decentralized Task Allocation for Dynamic Environments," Master's thesis, Massachusetts Institute of Technology, January 2012.
- [5] L. B. Johnson, H.-L. Choi, S. S. Ponda, and J. P. How, "Allowing non-submodular score functions in distributed task allocation," in *IEEE Conference on Decision and Control (CDC)*, Dec 2012 (submitted).
- [6] S. Zhu and M. Fukushima, "Worst-case conditional value-at-risk with application to robust portfolio management," *Operations research*, vol. 57, no. 5, pp. 1155–1168, 2009.
- [7] H. Balakrishnan and B. Chandran, "Algorithms for Scheduling Runway Operations under Constrained Position Shifting," *Operations Research*, vol. 58, no. 6, 2010.
- [8] D. Bertsimas, "Probabilistic combinatorial optimization problems," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1988.
- [9] D. P. Bertsekas, "Dynamic programming and optimal control," 1995.
- [10] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Operations Research Letters*, vol. 25, no. 1, pp. 1–14, 1999.
- [11] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007.
- [12] D. Bertsimas, D. Brown, and C. Caramanis, "Theory and applications of robust optimization," *Arxiv preprint arXiv:1010.5445*, 2010.
- [13] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, pp. 769–805, 1998.
- [14] W. Chen, M. Sim, J. Sun, and C. Teo, "From cvar to uncertainty set: Implications in joint chance constrained optimization," *Operations research*, vol. 58, no. 2, pp. 470–485, 2010.
- [15] D. Bertsimas and D. Brown, "Constructing uncertainty sets for robust linear optimization," *Operations research*, vol. 57, no. 6, pp. 1483–1495, 2009.
- [16] E. Delage and S. Mannor, "Percentile optimization for markov decision processes with parameter uncertainty," *Operations research*, vol. 58, no. 1, pp. 203–213, 2010.
- [17] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," 2009.